Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 36 (2014) 168 – 174

Procedia Computer Science

Complex Adaptive Systems, Publication 4
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2014- Philadelphia, PA

Enhancing a Rule-Based Event Coder with Semantic Vectors

Jinhong K. Guo, David Van Brackle and Martin O. Hofmann

3 Executive Campus, Suite 600, Cherry Hill, NJ 08002, USA

Abstract

Rule based systems have achieved success in applications such as information retrieval and Natural Language Processing. However, due to the rigidity of pattern matching, these systems typically require a large number of rules to adequately cover the variations of expression in unstructured text. Consequently, knowledge engineering for a new domain and knowledge maintenance for a fielded system are labor intensive and expensive. In this paper, we present our research on enhancing a rule-based event coding system by relaxing the rigidity of pattern matching with a technique that formulates and matches patterns of the semantics of words instead of literal words. Our technique pairs literal words with semantic vectors that accumulate word meaning from the context of use of the word found in dictionaries, ontologies, and domain corpora. Our method improves the speed, accuracy, and coverage of the event coding algorithm without additional knowledge engineering effort. Operating on semantics instead of syntax, the improved system eases the workload of human analysts who screen input text for critical events. Our algorithms are based on high-dimensional distributed representations, and their effectiveness and versatility derive from the unintuitive properties of such representations---from the mathematical properties of high-dimensional spaces. Our current implementation encodes words, phrases, and rule patterns as semantic vectors using WordNet, We have started experimental evaluation using a large newswire dataset.

Peer-review under responsibility of scientific committee of Missouri University of Science and Technology

Keywords: NLP, semantic vectors, Random Indexing, event coding, rule-based system, machine learning

1. Event Coding, Rule Based Systems and Limitations

Political event data have long been used in the quantitative study of international politics, dating back to the early efforts of Edward Azar's COPDAB [1] and Charles McClelland's WEIS [7] as well as a variety of more specialized efforts such as Leng's BCOW [6]. By the late 1980s, the NSF-funded *Data Development in International Relations* project [8] had identified event data as the second most common form of data—behind the various Correlates of War data sets— used in quantitative studies. The 1990s saw the development of two practical automated event data coding systems, the NSF-funded KEDS (http://eventdata. psu.edu; [3, 12, 14]) and the proprietary VRA-Reader (http://vranet.com; [5, 11]), and in the 2000s, the development of two new political event coding ontologies, CAMEO [15] and IDEA [2,11], designed for implementation in automated coding systems. A summary of the current status of political event projects, as well as detailed discussions of some of these, can be found in [4, 13].

Peer-review under responsibility of scientific committee of Missouri University of Science and Technology
doi:10.1016/j.procs.2014.09.074

While these efforts had built a substantial foundation for event data—by the mid-2000s, virtually all refereed articles in political science journal used machine-coded, rather than human-coded, event data—the overall development of new technology remained relatively small. This situation changed with the DARPA-funded Integrated Conflict Early Warning System (ICEWS) [9, 10], which utilized event data development coded with automated methods. The key difference between the ICEWS event data coding efforts and those of earlier NSF-funded efforts was the scale. The later phases of ICEWS [9] moved to near-real-time, global event data production and the scale of this coding effort increased even further, now covering 175 countries and nearly 20-million stories [16].

There are two main approaches to event coding, statistical and rule-based. Both methods have advantages and disadvantages. A statistical method may have some advantage in accuracy; however, it needs labelled training data which is labor intensive, and for any change, it needs to go through retraining of the system.

Jabari is a rule-based event coder that was built from Textual Analysis by Augmented Replacement Instructions http://web.ku.edu/~keds/software.dir/tabari.html (TABARI). Jabari was one of the event coders for the Integrated Crisis Early Warning System (ICEWS) and has been applied to many other projects. In comparison to statistical methods, to transfer Jabari into a relatively new domain, a set of new rules needs to be engineered based on domain knowledge. Empirically, a relatively small number of rules (in the tens) can find about 50 percent of the events. However, improvements beyond this level of performance require disproportionally more rules. Perpetually augmenting and maintaining the rule base is a challenge. Additionally, the brittleness of the pattern matching employed by Jabari affects the precision and recall of the processing results. The following example illustrates the problem discussed. Jabari event rules have a Main Verb, and a series of patterns using that verb, which Jabari matches word-by-word, e.g.

```
ABANDON
- * HEADQUARTERS  [0874]
- * STRONGHOLD  [0874]
- $ * OUTPOST IN_ +  [0874]
- $ CREW * SHIP_ +  [---]
- * RE_TRIAL  [---]
- * ATTEMPT  [080]
- * PLAN_ [---]
- * PARTY  [160]
- * TRUCE  [196]
- * ROLE  [160]
```

In this example, due to the exact pattern matching employed by Jabari to ensure better accuracy, the best practice is to list all the possible patterns known at development time that include the verb ABANDON to cover as many of the relevant events as possible.

This paper describes our research aimed at injecting statistical learning into Jabari's pure rule-based system, reducing the brittleness of the pattern match and the never-ending cost of rule building. Our idea is to represent words, phrases and patterns by their semantic meanings to eliminate the need to keep adding all the possible verb patterns.

2. Semantic Jabari

In this section, we discuss the details of our enhanced system, called Semantic Jabari. A key innovation is to represent words and phrases, therefore, patterns, using a semantic vector representation [17][18]. A semantic vector is a high-dimensional vector that captures the meaning of a language expression at different granularities, such as a word, phrase, or query, or a whole news article, story, or message, so that expressions with similar meaning are encoded by vectors that are similar to each other in vector space. We chose a highly scalable technique called Random Indexing [19] as the basis for generating semantic vectors. Random Index vectors are constructed based on the observation that words with similar meanings tend to occur in similar contexts. Word sequence information, too, can be encoded into high dimensional fixed-length vectors using permutation of coordinates [20], helping to disambiguate the meaning of phrases that are sensitive to word order.

We encode each verb defined in the verb dictionary of Jabari and each word in its associated patterns with a semantic vector, using word context from the Jabari dictionaries and from generic dictionaries, such as WordNet. The pattern matching between the predefined event patterns and the text to be processed becomes a similarity comparison between two vectors for each word, allowing approximate matching and, thus, reducing the brittleness of the rule-based system. Currently, if a rule is built for "subject requests object", an event would not be extracted from sentence "A requires B" unless another rule is built for the verb "require". The semantic meaning for “request” and “require” can be encoded from WordNet. While the semantic vectors for “require” and “request” are not an exact match, they are highly similar. Extending Jabari with semantic pattern matching will preserve its current advantages of a short training time and will ease the effort required to add new event types. The added capability will increase the recall rate by matching semantically similar verbs and words that are not in the Jabari dictionary. It can also increase accuracy by selecting the best matching patterns from among competing alternatives. Pure rule-based matching, on the other hand, has no basis for selecting the best match and tends to select the first rule that matches. When fielded, we let the user control the degree of vector similarity necessary to declare a match. Thus, the user can tune the system for higher precision or recall. Selecting a similarity measure of 1.0 requires a perfect match and reverts the behavior of Semantic Jabari to that of the current Jabari.

2.1. Semantic Representation

Current statistical algorithms (e.g., LSA, probabilistic Latent Semantic Analysis (pLSA)[21], and Latent Dirichlet Allocation (LDA)[22]) model the statistics of word co-occurrence or word sequences (as in Conditional Random Fields (CRF)[23]), and have led to tremendous progress in machine translation and topic discovery, but these methods largely operate on surface features, ignoring syntactic or semantic structure beyond part-of-speech tagging. Scaling using these conventional methods can also be a problem. In contrast, our approach makes use of additional linguistic information, and will overcome conventional methods’ lack of semantics and scaling.

Semantic vectors capture the meanings of words based on the contexts in which they occur. Our method allows a rich variety of contexts, including neighboring words in text, larger contexts such as paragraphs, and "relational contexts" as given by WordNet (http://wordnet.princeton.edu). The resulting semantic vectors for "similar" words are close to each other, but the nature of the similarity depends on the context used to derive the vectors, and includes both substitutional similarity ("doctor" and "nurse") and relatedness ("doctor" and "hospital").

Our algorithm incorporates word vectorization through a novel high dimensional random vector technique described in Kanerva [24]. The use of random indexing as a means of storing and comparing text statements has great potential. There are many ways to compare text. The most common way currently is to use a sliding window around a word and collect the words used with it to help define this word and find the similarity in its use with other words. A second approach is to use all the words in a statement to build a semantic vector, which captures the meaning of the word in that context. Capturing this data creates a very large workspace, which is difficult to store and process in real time. While many applications do not require real time processing, it is important in many applications, such as social network analysis. The storage space required to handle a large text corpus is also a concern with many methods. Random Indexing using sparse vectors is a good approach to overcoming both of these concerns.

A key benefit of semantic vectors is the substantial dimensionality reduction from projecting very large vocabularies onto, for our example, an eight thousand dimensional space of semantic vectors. This allows good generalization to rarely seen words, which "borrow strength" from their more frequent neighbors. It allows incorporation of larger contexts and it speeds computation over the more memory intensive n-gram language models. Our semantic vector model representations will scale to very large training corpora and will enable better coverage of rare terms and higher certainty for normal text patterns as well as easy inclusion of additional information to the current baseline. This method is also language agnostic and little effort is required to apply them to new foreign languages.

2.2. Encoding Word Semantic Vector Using WordNet

Each word in WordNet contains a definition and a set of synonyms that represent the same concept. For example, the concept home can also be expressed as dwelling, domicile, abode, or habitation, and many others. These words are known as a synset in WordNet. Each word is also associated with a definition, as well as a gloss. A gloss is a partial sentence that contains the meaning of the work in a certain sense; for example, a gloss of home is “deliver the package to my home”. The definition for this first sense of home is “where you live at a particular time”. Each Random Indexing (RI) representation of one definition and gloss became a first order vector for each word or concept. The second order vector for a word is created by combining its first order vector with the first order vector of the other concepts.

The algorithm for generating a semantic vector for words is as following.

1. Initiate a RI vector for each word
    a. Start with a zero vector of length *d* (e.g., *d=8000*)
    b. Turn *k* (e.g., *k=10*) 0s into *+1*s at random
    c. Turn *k* (e.g., *k=10*) 0s into *-1*s at random
2. Build semantic vector for word *i*
    a. Embed gloss for word *i* by "adding" the RI for each of the words in gloss to the RI of word *i*
    b. Similarly, embed definition of word *i*
    c. After all embedding, change any vector element >0 to 1, <0 to -1, so that the final vector consists only of -1s, 0s or 1s.

Semantic vectors can be compared for similarity by a distance measure. In our experiments, we use the following measure: Count the number of vector positions where both vectors are 1, or both vectors are -1. Divide this count by the square root of [the number of non-zero terms in the first vector, times the number of non-zero terms of the second vector]. This measure of similarity is very similar to the Cosine measure of similarity. The Cosine measure uses a penalty of -1 for a mismatch, where one vector has a 1 and the other has a -1. The Cosine measure makes the most sense when antonym vectors are embedded and subtracted, which we do not do.

In the experiments, we removed stop words (short function words, such as the, is, at, which, etc.) and find the lemmas (grouping different inflected forms of a word) of each word when generate the semantic vectors.

2.3. Enhancing Original Jabari with Semantic Similarities

We integrated the RI based semantic vectors for the words with the Jabari software. The following experiment serves as a feasibility study for further research on Semantic Jabari. For the experiment, we implement a word-by-word match between the verb rules and the sentences in question.

Jabari event extraction operates on each sentence. It uses OpenNLP to produce a parse tree and tokenize the sentence, then uses pattern matching to identify potential actors, agents and verbs based on the dictionaries. Jabari tries to match the available verb patterns to the verb phrase in the sentence, using the parse tree's Part-of-Speech analysis as a guide. On a match, the parse tree is then used to isolate the Source and Target of the event. The difference between Jabari and Semantic Jabari is that, when matching the verb phrase, instead of an exact verb pattern match, Semantic Jabari performs a semantic match using semantic vector similarities between the corresponding words.

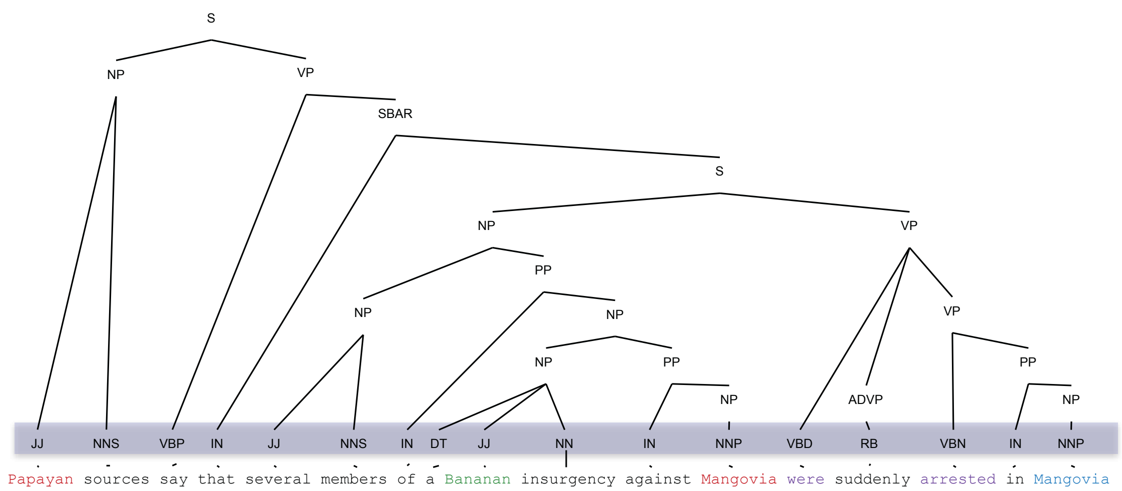

Figure 1 Example of a sentence and its parse tree.

Figure 1 shows an example that was processed into a parse tree. This sentence is in Passive Voice, as indicated by the words in Purple. The correct Target is in Green, correct Source is in Blue, other Actors are in Red. The Jabari rule for this is the main verb "ARREST". Jabari will treat the main verb as a pattern unto itself. It also recognizes passive voice.

3. Experiments

Our preliminary experiment was targeted at demonstrating that Semantic Jabari improves recall over purely rule-based Jabari without sacrificing accuracy. The ICEWS program has conducted human evaluations on large-scale experiments, and the evaluation process proved to be very tedious and resource consuming. We planned to take advantage of ICEWS results as much as possible. We set up our experiments using the same ICEWS dataset. We observed that for the events that were extracted and coded by both original Jabari and Semantic Jabari, Semantic Jabari produced similar if not better results. Therefore, we focused our evaluation on event coding of the news stories where original Jabari failed to extract any events due to the brittleness of the event patterns.

3.1. Dataset

Semantic Jabari was tested on a corpus of data that has been used in ICEWS. It consists of 615,000 news stories from 2008 and 2009, with the first four sentences of each story. The stories come from a multitude of sources.

3.2. Experiment Setup and Results

We performed two experiments on the ICEWS dataset. As in original Jabari, we used OpenNLP to produce a parse tree and tokenize the sentence, then used pattern matching to identify potential actors, agents and verbs based on the dictionaries. The difference between Semantic Jabari with the original Jabari is that instead of requiring an exact match of the words in the verb pattern, we semantically match the meaning representation of the words as represented by semantic vectors.

Figure 2 shows an example that illustrates how Semantic Jabari relaxes the brittleness of the original Jabari via semantic matching. In this example, the applicable rule implies that if a sentence matches "EXCHANGE VIEW", it will be coded as a "consult" event with code [040]. Since the original Jabari has no rule that corresponds to "EXCHANGE IDEAS", the event that should be extracted from this sentence is missed. Since the semantic meanings of the two expressions are similar, Semantic Jabari was able to extract and code the corresponding event.

Example Sentence:

```
Nearly 100 scientists, environmentalists,
politicians and UN officials exchanged ideas at a
whaling symposium arranged by the Pew Charitable
Trusts, a non-governmental US research institute.
```

Applicable Rule:

```
EXCHANGE
 - % * VIEW [040]
```

Figure 2 Example sentence and the verb rule it matches

The example in Figure 2 illustrates how difficult and time consuming is to write rules for every expression for a verb. It is also difficult to write rules for all the verbs that may appear, even with a specific domain. In another example, "Pyongyang, January 31 (KCNA) - The US moves to isolate and stifle Iran are going awry." There was a rule for the verb "STRANGLE", but none for "STIFLE" which is semantically similar.

We conducted two sets of experiments. In the first set, we generated one semantic vector for each word, combining all of the word senses available in WordNet. We used a similarity measure threshold of 0.3 (similarity ranged from 0 to 1) for a match. Semantic Jabari extracted 40 percent more events than original Jabari with 50 percent accuracy for the additional events that were coded.

We recognize that a word can have very different meaning with different senses and that the verb plays an essential role in event coding. In previous research [17,18], experiments have shown that the best result will come from keeping the senses totally separate as well as by having more words, definitions and glosses in the dictionary for the representation of each word. In the current implementation, we do not know the exact Part-of-Speech of each word in the sentence with enough accuracy, but we know which word is a verb.

In our next experiment, we generated two semantic vectors for each word, one combining all senses of the word as a verb, the other combining all other senses of the word (noun, adjective, adverb). We used a tighter similarity measure threshold of 0.5. We then used the semantic vector of the verb sense for the main verb, and used the semantic vector of all other senses for all other words in a pattern. Under these conditions, Semantic Jabari extracted over 4 percent more events than original Jabari, with 73 percent accuracy.

4. Conclusions

Random indexing captures context-based semantics in high-dimensional vectors to facilitate semantic comparison. The experiments presented above showed promise in using a semantic vector representation to enhance rule based systems. Higher recall obviates the need for adding rules to cover the events that the current Jabari rule set misses. Using semantic similarity, we can augment the rules automatically, hence reducing the cost of perpetual rule engineering. Unlike rule matching which can be affected by the order of rules in the knowledge base, semantic similarity matching matches to the best verb pattern, thus, enhancing precision while increasing recall of the system. Despite these enhancements, we maintained Jabari's capability of fast adaptation to changes in the application domain.

Our semantic language model embeds word context and relations, e.g., synonym, antonym, PoS etc., from a lexicon, such as WordNet, It is possible to further bias the semantic meaning to a specific domain by word context and relationships from an unlabeled training corpus.

In our preliminary experiments, similarity matching is performed based on word to word comparisons between the verb pattern defined in the dictionary with the verbs and nouns parsed from a sentence. In the RI approach, the structure and ordering of a language will be captured through permutation and projection of these random indexing vectors [20], flattening a sequence of vectors `a b c d ...' into a single vector. Other researchers have proposed different ways of semantic compositions [25]. These will allow us to encode the rules and the parse tree that resulted from the sentence in the text being processed into one semantic vector. Context information and syntactic structure can also be encoded in that vector. Instead of comparing word semantic vectors word-by-word, we will be comparing words, patterns and meanings of the sentences in the context they appear.

Acknowledgements

We thank Dr. Rosemary Paradise at Lockheed Martin IS&GS for the discussion and support in the research of semantic equivalence. We also thank Sara Jacob for her help in taking the tedious task of evaluating the experiment results.

References

1. Azar EE (1980) The conflict and peace data bank (COPDAB) project. J Confl Resolut 24: 143–152
2. Bond D, Bond J, Oh C, Jenkins JC, Taylor CL (2003) Integrated data for events analysis (IDEA): An event typology for automated events data development. J Peace Res 40(6): 733–745
3. Gerner DJ, Schrodt PA, Francisco RA, Weddle JL (1994) The machine coding of events from regional and international sources. Int Stud Q 38:91–119
4. Gleditsch NP (2012) Special issue: event data in the study of conflict. Int Interact 38(4): 375–569
5. King G, Lowe W (2004) An automated information extraction tool for international conflict data with performance as good as human coders: A rare events evaluation design. Int Organ 57(3):617–642
6. Leng RJ (1987) Behavioral correlates of war, 1816–1975. (ICPSR 8606). Inter-University Consortium for Political and Social Research, Ann Arbor
7. McClelland CA (1976) World event/interaction survey codebook (ICPSR 5211). Inter- University Consortium for Political and Social Research, Ann Arbor
8. Merritt RL, Muncaster RG, Zinnes DA (eds) (1993) International event data developments: DDIR phase II. University of Michigan Press, Ann Arbor
9. O'Brien S (2012) A multi-method approach for near real time conflict and crisis early warning. In: Subrahmanian V (ed) Handbook on computational approaches to counterterror- ism. Springer, New York
10. O'Brien SP (2010) Crisis early warning and decision support: contemporary approaches and thoughts on future research. Int Stud Rev 12(1):87–104
11. Petroff V, Bond J, Bond D (2012) Using hidden Markov models to predict terror before it hits (again). In: Subrahmanian V (ed) Handbook on computational approaches to counterterrorism. Springer, New York
12. Schrodt PA (2006) Twenty years of the Kansas event data system project. Political Methodol 14(1):2–8
13. Schrodt PA (2012) Precedents, progress and prospects in political event data. Int Interact 38(5):546–569
14. Schrodt PA, Gerner DJ (1994) Validity assessment of a machine-coded event data set for the Middle East, 1982–1992. Am J Political Sci 38:825–854
15. Schrodt PA, Gerner DJ, Yilmaz O¨ (2009) Conflict and mediation event observations (CAMEO): an event data framework for a post Cold War world. In: Bercovitch J, Gartner S (eds) International conflict mediation: new approaches and findings. Routledge, New York
16. Shilliday A, Lautenschlager J (2012) Data for a global icews and ongoing research. In: 2nd international conference on cross-cultural decision making: focus 2012, San Francisco, CA
17. Paradis, P.D.,(2013) Guo, J.K., Moulton, J., Cameron, D., Kanerva, P., Finding Semantic Equivalence of Text Using Random Index Vectors, Complex Adaptive Systems 2013
18. Paradis, R.D.(2013a), Moulton, J., Guo, J.K., Kanerva, P., Using Random Index Vectors for Semantic Equivalence of Text using WordNet, poster presentation, IJCNN 2013
19. Kanerva, P., Kristoferson, J., and Holst, A. (2000). Random indexing of text samples for Latent Semantic Analysis. In L.R. Gleitman and A.K. Josh (eds.), Proc. 22nd Annual Conference of the Cognitive Science Society (Philadelphia), p. 1036. Mahwah, New Jersey: Erlbaum.
20. Sahlgren, M., Holst, A. and Kanerva, P., Permutations as a Means to En-code Order in Word Space. Proceedings of the 30th Annual Meeting of the Cognitive Science So-ciety, Washington, DC. Pp.1300-1305. 2008
21. Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, *42*(1-2), 177-196.
22. Blei, D.M., Ng, A.Y., & Jordan, M.I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, *3*, 993-1022.
23. Lafferty, J., McCallum, A., & Pereira, F.C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of International Conference on Machine Learning (ICML) 2001*.
24. Kanerva, P. (2009). Hyperdimensional Computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation* 1(2):139-159. http://redwood.berkeley.edu/pkanerva/papers/kanerva09-hyperdimensional.pdf
25. Mitchell, J. and Lapata, M., Vector-based Models of Semantic Composition, Proceedings of ACL-08, pp. 236-244